



**HAL**  
open science

# RealPi2dDIC: A low-cost and open-source approach to In-situ 2D Digital Image Correlation (DIC) applications

Partha Pratim Das, Muthu Ram Prabhu Elenchezian, Vamsee Vadlamudi,  
Kenneth Reifsnider, Damien André, Rassel Raihan

## ► To cite this version:

Partha Pratim Das, Muthu Ram Prabhu Elenchezian, Vamsee Vadlamudi, Kenneth Reifsnider, Damien André, et al.. RealPi2dDIC: A low-cost and open-source approach to In-situ 2D Digital Image Correlation (DIC) applications. SoftwareX, 2024, 27, pp.101864. 10.1016/j.softx.2024.101864 . hal-04863813

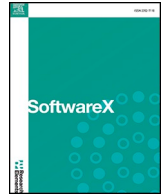
**HAL Id: hal-04863813**

**<https://hal.science/hal-04863813v1>**

Submitted on 4 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# RealPi2dDIC: A low-cost and open-source approach to In-situ 2D Digital Image Correlation (DIC) applications

Partha Pratim Das<sup>a,b,\*</sup>, Muthu Ram Prabhu Elenchezian<sup>a,b</sup>, Vamsee Vadlamudi<sup>b</sup>,  
Kenneth Reifsnider<sup>b</sup>, Damien André<sup>c</sup>, Russel Raihan<sup>a,\*</sup>

<sup>a</sup> Department of Mechanical and Aerospace Engineering, University of Texas at Arlington, Arlington, TX 76019, USA

<sup>b</sup> Institute for Predictive Performance Methodologies, University of Texas at Arlington Research Institute, Fort Worth, TX 76118, USA

<sup>c</sup> Univ. Limoges, IRCER, UMR CNRS 7315, 87000 Limoges, France

## ARTICLE INFO

### Keywords

Digital Image Correlation (DIC)  
Lucas-Kanade Algorithm  
Experimental Mechanics  
In-situ strain measurement

## ABSTRACT

In this current work an open-source 2D Digital Image Correlation (DIC) tool, RealPi2dDIC, is presented to monitor in-situ full field deformation and strain responses of structures during loading. This software is coded in Python3 and uses Raspberry Pi as the computation and Pi camera as the imaging device. This low-cost approach to in-situ DIC analysis enables its application in a broad field of research and industry problems where 2D strain field tensor is one of the principal interests of study. This code is provided with an interactive user interface (UI) making it user friendly and easy-to-use. Furthermore, RealPi2dDIC offers a simple software architecture which can be adopted, modified and extended to suit users' purpose.

## 1. Motivation and significance

When it comes to the development and maintenance of different structures under loading (mechanical and thermal), strain and deformation are critical parameters to consider. In experimental solid mechanics, a single scalar value of strain is measured using pointwise strain gauges such as electrical resistance strain gauge, extensometers and using cross head movements of the testing equipment. But it considers the strain as homogeneous which does not represent the true nature of strain. Hence, these methods have been found to be imperfect in recording the true characteristics of materials under testing [1,2]. To acquire local deformation information for inhomogeneous deformation fields, several non-contact optical methods, including speckle interferometry, moiré interferometry, the grid method and digital image correlation (DIC) are developed and implemented [3–5]. Through the last few decades, DIC has been developed significantly into an effective and flexible tool for surface measurement from nanoscale to macroscopic level compared to the other techniques [6]. In DIC technique, a series of images of an object under ongoing deformation are captured and compared with a non-deformed reference image. Typically, a speckle-like pattern is applied on the region of interest (ROI) of the specimen which is used to correlate the image coordinates of an image

and a reference image using a proper optimization algorithm [7]. Thenceforth, from the coordinate mapping, full field strain is computed. With the continuous advancement of DIC technique, a number of algorithms has been developed for correlating images, including the most common ones- finite element-based DIC method (FE-based global DIC) [8], subset based Newton-Raphson (NR) algorithm [9] (Table 1).

Over the years, DIC has been used in various scientific fields as a post processing tool for strain measurement. However, in real-world applications in-situ monitoring of strain development is much desired. Continuous development of the NR algorithm ensured the employment of this to acquire highly accurate DIC measurements [10,11], but one disadvantage of NR algorithm, however, is the requirement of significant computational time [12]. Due to this limitation, application of DIC in real-time displacement and strain measurements is still an ongoing research effort. In the recent years, algorithmic development has been explored to minimize the computational cost for DIC measurements. Tao and Xia developed a real-time measurement system which required a high-end CCD camera and two separate computers to acquire and process data to increase the computation speed [13]. Wu et al. developed a new integer pixel search scheme [14] which ensured a higher computational speed for real-time application of DIC. The algorithm was coded using C++ language and depends on the computational device

DOI of original article: <https://doi.org/10.1016/j.softx.2020.100645>.

\* Corresponding authors.

E-mail addresses: [parthapratim.das@mavs.uta.edu](mailto:parthapratim.das@mavs.uta.edu) (P.P. Das), [mrdassel.raihan@uta.edu](mailto:mrdassel.raihan@uta.edu) (R. Raihan).

<https://doi.org/10.1016/j.softx.2024.101864>

Received 6 December 2022; Received in revised form 26 June 2024; Accepted 22 August 2024

Available online 7 September 2024

2352-7110/© 2024 The Author(s). Published by Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

**Table 1**  
Code metadata.

Code Metadata	
Current code version	0.0.1
Permanent link to code/repository used of this code version	<a href="https://github.com/utaresearch/RealPi2dDIC">https://github.com/utaresearch/RealPi2dDIC</a>
Code Ocean compute capsule	n/a
Legal Code License	GNU-GPLv3
Code versioning system used	none
Software code languages, tools, and services used	Python3, opencv, numpy, scipy, picamera, matplotlib
Compilation requirements, operating environments & dependencies	Raspberry Pi, Raspbian OS, Pi camera module v2
If available Link to developer documentation/manual	<a href="https://utaresearch.github.io/RealPi2dDIC/docs.html">https://utaresearch.github.io/RealPi2dDIC/docs.html</a> <a href="https://utaresearch.github.io/RealPi2dDIC/sop.html">https://utaresearch.github.io/RealPi2dDIC/sop.html</a>
Support email for questions	parthapratim.das@mavs.uta.edu

critically. Although these methods exhibit high accuracy in results, it is still a challenge to balance between mobility, simplicity, accuracy and cost when practical applications are considered.

Commercial DIC software such as *VIC-2D*, *GOM Correlate*, *LaVision StrainMaster* (including own imaging system), *MatchID-2D* and open source software like *nCorr* [15], *ALDIC* [16], *DICe* [17], *Py2DIC* [18], *pydic* [19],  $\mu$ *DIC* [20], *YADIC* [21], *Ufreckles* [22] are now available. Open source software is more preferred in academia as they accommodate the flexibility of modifications. However, except *Py2DIC* and  $\mu$ *DIC*, the other codes are either depended on the commercial software, or formulated as scripts with very limited documentation [20]. Moreover, each of the software can only be used as a post processing tool for deformation and strain measurements using DIC. *VIC-2D Gage* and *Dantec Systems DIC Solutions* provide some DIC systems for in-situ monitoring of deformation and strain, but they are only available commercially.

In this prospect, RealPi2dDIC brings a solution to in-situ monitoring of 2D full field strain through a very low-cost approach. In this system, a Pi camera module v2 is used as the image capturing device and a Raspberry Pi model 3B+ is used as the computing device which has reduced the cost of equipment significantly while increasing mobility and simplicity. It has been programmed in one of the most accessible and popular programming languages- Python3. To optimize the overall processing speed, a very well established optical flow technique - Lucas-Kanade (LK) algorithm [23] is used which works based on subset based NR algorithm at a faster speed [24]. To incorporate the LK method in the program, the open source computer vision library for Python3, *OpenCV* [25] has been used which also increases the processing speed. For fast array operation and interpolation, *NumPy* [26] and *SciPy* [27] libraries are used in this software. All these optimization techniques combinedly make RealPi2dDIC an effective system to monitor in-situ strain over a planar body in both indoor applications, e.g. material characterization tests in a laboratory setup, etc. and outdoor applications, e.g. beams and bridge deflection monitoring, etc. The user can easily implement this hardware-software system in desired applications following the experimental procedure explained in the examples.

## 2. Software description

RealPi2dDIC is a software consisting of a few high-level classes and functions which is used for capturing images utilizing a Pi camera module, generating data for image correlation, interpolation and data post-processing. Given the flexibility of a Pi camera module, this software makes it usable in different lighting conditions by controlling the capture parameters. The initial correlation approach includes the independent analysis of points using local subsets (termed as ‘‘correlation window’’) about each point within a Region of Interest (ROI) of a reference image captured in the beginning with no deformation. The

ROI is selected by the user through an interactive UI. The displacements of the points are computed by mapping the subsets in the reference image with corresponding subsets in a subsequent image after deformation. This system offers subpixel accuracy aided by interpolation [18] of the local displacements as well raw displacements (without any interpolation). Using the results from the interpolated or raw displacement, strain field is calculated and prepared for post processing, and visualization through a contour plot over the ROI. The raw data generated by this system also offers the flexibility of using different other tools for post-processing.

### 2.1. Software architecture

RealPi2dDIC is coded in Python 3.6 using a few external packages (principally *OpenCV*, *NumPy*, *SciPy*, *Picamera* [28], *Matplotlib* [29]). This code is motivated by *pydic* [19] incorporating additional features (i.e. real time data acquisition and processing, use of low-cost embedded device for enhanced application). The choice of *pydic*, as the core architecture for RealPi2dDIC was motivated by several aspects, including a high level of versatility and simplicity, dependency on external packages (i.e. *OpenCV*, *Numpy*, *Scipy*, *Matplotlib*) that makes the software compatible in terms of computational performances with the RaspberryPi device. In addition, *pydic* offers several original numerical treatments able to improve the accuracy of results in terms of displacement and strain fields measurements. These treatments are based on removing rigid body movements algorithm and spline smoothing method for reducing noises induced by image acquisition artifact such as optical problems, kinematic chain problems, optical sensor sensitivity, etc. RealPi2dDIC uses the same global architecture as *pydic* with additional steps and components for real time data acquisition and preprocessing. A simple flow of processes occurring in an instance of RealPi2dDIC is shown in Fig. 1.

Being highly modular, this software consists of multiple processes running concurrently:

- *grid*: A core class that contains data processed by DIC measurement with a collection of functions for initiating required interactive UI, performing image correlation, and processing and exporting raw data.
- *DIC\_Run*: A simple interface function with user which calls most of the important functions including functions used for capturing images, defining ROI, writing raw image data and correlated data in a text file. All the events of this software are wrapped in a loop in *DIC\_Run*.
- *raw\_Data\_Process*: It calls a few functions used for reading raw image data and correlated data from generated txt file, followed by performing data interpolation and exporting raw data
- *Plot*: A class which includes all the required operations for visualization of the computed strain field

RealPi2dDIC software architecture including the main components and their functions are demonstrated in Fig. 2.

### 2.2. Software functionalities

The main functionalities of RealPi2dDIC are explained in the following:

#### 2.2.1. Flexible image capture operation

RealPi2dDIC uses a Raspberry Pi camera module v2 as its image capturing device for DIC. Pi camera v2 is a very lightweight camera module weighing only 3 g. It has a Sony IMX219 sensor with 3280 × 2464 pixels resolution and 3.04 mm focal length. It allows the user to control the image capturing parameters such as brightness, contrast, exposure compensation, ISO, white balance and exposure mode. For a common exposure of the images captured, ‘beach’ exposure mode is

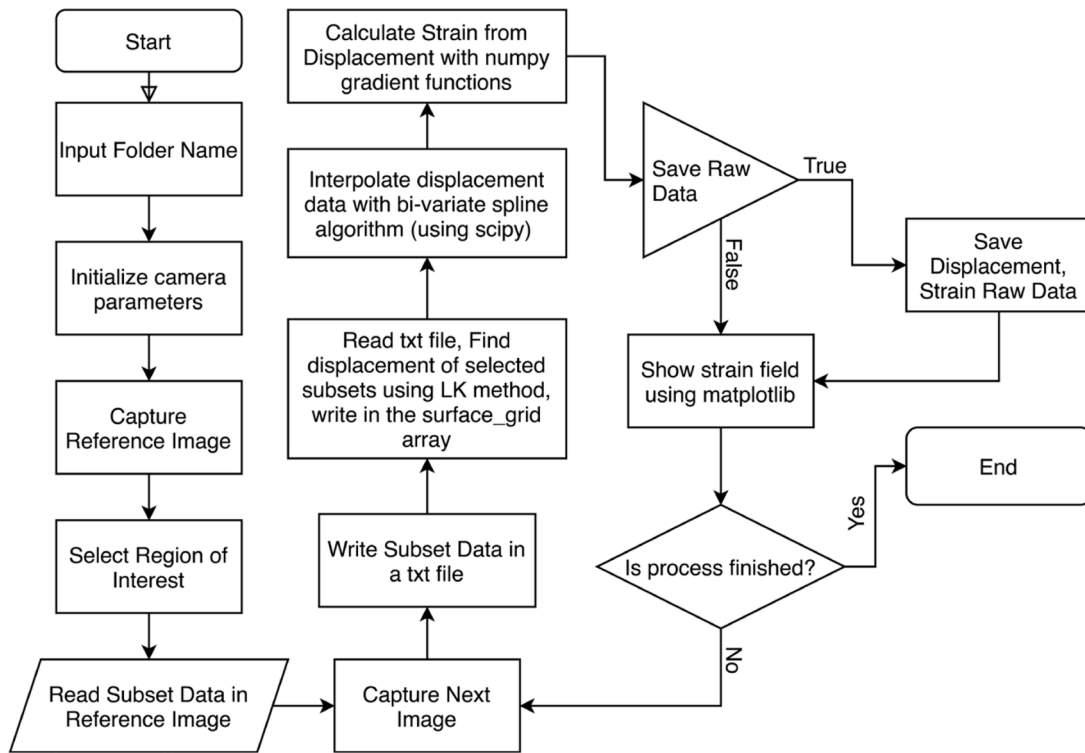


Fig. 1. Flow chart of RealPi2dDIC for in-situ strain monitoring of a structure under loading.

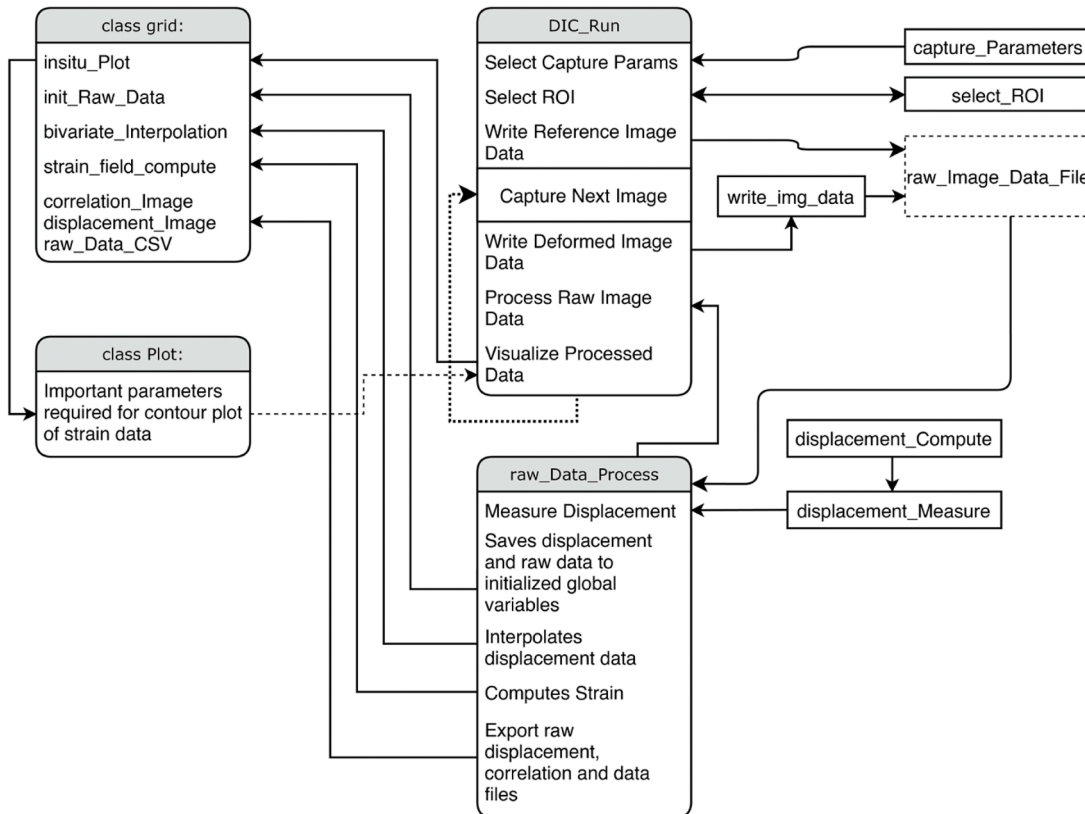


Fig. 2. A full pipeline showing the main components of RealPi2dDIC and their functions.

preferred. Proper camera white balance should be selected based on the color temperature of the light source in use. Different options for exposure mode and white balance are available and can be found in the

*picamera* module's documentation [28]. The brightness, contrast, exposure compensation and ISO can be controlled using an interactive UI (Fig. 3. (a)) coded with *OpenCV* which allows a greater flexibility in

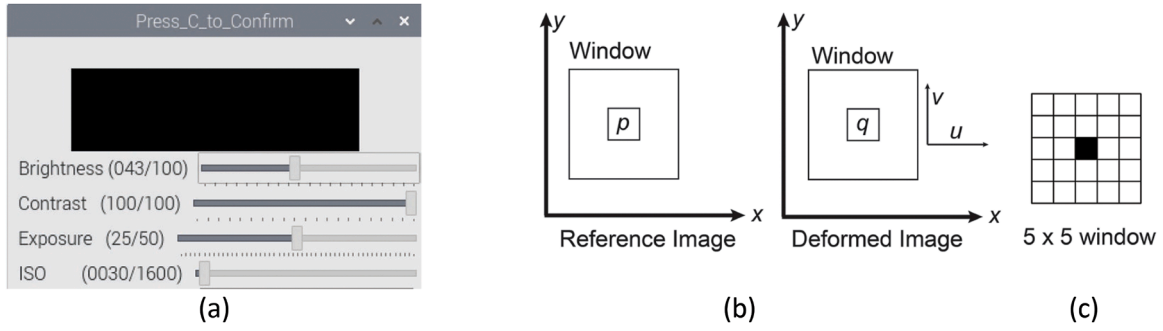


Fig. 3. (a) UI for selecting Image capture parameters, (b) Schematic diagram of the reference image and deformed image, (c) central pixel and correlation window.

selecting these key parameters based on the lighting arrangements. The software architecture of RealPi2dDIC can also be modified to incorporate different source camera and image formats.

### 2.2.2. Image correlation

This is the main function of this software where the correlation between a deformed image and reference image is done using LK optical flow algorithm [23]. This is a well-established method, used widely in the field of computer vision [30]. LK algorithm makes the best calculation of the displacement of a neighborhood of pixels, assuming locally constant motion of the pixels.

Grey-scale intensity measurement is the main tool in LK algorithm. Fig. 3. (b) represents a square window through which the intensity of a single pixel  $p$  can be detected. In the consecutive image, due to deformation the intensity of the pixel has been changed to  $q$ . This algorithm assumes that a displacement of the pixel has occurred in both the  $x$  and  $y$  directions as the new intensity is visible through the square window. It leads to the general formation of the Eq. (1),

$$I_x(x,y) \cdot u + I_y(x,y) \cdot v = -I_t(x,y) \quad (1)$$

where,  $u$  and  $v$  are the displacement of the pixel in two directions,  $I_x$  and  $I_y$  are the intensity per pixel at  $(x,y)$  coordinate,  $I_t$  is the local difference in intensity ( $p-q$ ). In real-world conditions, Eq. (1) cannot be solved directly as there are more variables than equations. It is, therefore in LK algorithm, assumed that the neighboring pixels has the same displacement  $(u,v)$ . For example, if a subset of  $5 \times 5$  pixels is considered, there would be 25 equations for 2 variables  $(u,v)$  (Eq. (2)).

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{x25} & I_{y25} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{t25} \end{bmatrix} \quad (2)$$

The overdetermined system in Eq. (2) is solved using Least Square Solution (LSQ) principle shown in Eq. (3):

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \quad (3)$$

Specifically, in RealPi2dDIC the algorithm is applied through the OpenCV function `calcOpticalFlowPyrLK` [25]. Along with implementing LK method it uses pyramid levels [31] for more computational accuracy of small and larger motions. In RealPi2dDIC, a pyramid level of maximum 10 is used. In the higher level of the pyramid, small motions are neglected, and large motions are scaled to small motions. While LK method is used to compute small displacements, the implementation of pyramid levels leverages the algorithm to detect large motions in real-world scenario [32]. In DIC analysis, the window size has substantial consequences on the precision and speed of displacement measurement which also depends on the type of speckle pattern. Pan et al. conducted a study on the window size selection for speckle patterns of different grayscale distributions [33]. For high contrast speckle

patterns, Pan et al. observed the least standard deviation (SD) for  $65 \times 65$  to  $70 \times 70$  pixels, henceforth, this parameter in this implementation has been kept to an optimum of  $65 \times 65$  pixels to have the least standard deviation and higher computation speed. However, the user can also change the subset size based on the type of speckle pattern inside the source code.

### 2.2.3. Displacement interpolation

In most DIC algorithms, displacement is computed in pixel increments, however, the test specimen is not supposed to move in pixel increments. It results in erroneous estimation accompanied by noise. To get rid of this issue, interpolation of the displacement field is required. RealPi2dDIC offers a displacement field computation both with and without interpolation. If the interpolation parameter in `raw_Data_Process` is selected as `bivar_Spline`, a bivariate B-spline function for surface fitting is implemented to flatten the displacements computed at each pixel of the points considered. Following the recommendations of Dierckx [34], a 5th order spline function is implemented using the `SciPy` library functions `scipy.interpolate.bisplrep` and `scipy.interpolate.bisplev` [27]. It handles the missing data, reduces noise and smoothens the displacement field with highest accuracy possible.

### 2.2.4. Full-field strain measurement and visualization

In RealPi2dDIC, full field strain is computed from the interpolated displacement field, implementing the finite strain theory as followed in multiple DIC applications [18,35,36]. The strain, therefore, is obtained using the Lagrangian finite strain tensor shown in Eq. (4).

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{xy} & \epsilon_{yy} \end{bmatrix} \quad (4a)$$

$$\epsilon_{xx} = \frac{\delta u}{\delta x} + \frac{1}{2} \left( \left( \frac{\delta u}{\delta x} \right)^2 + \left( \frac{\delta v}{\delta x} \right)^2 \right) \quad (4b)$$

$$\epsilon_{yy} = \frac{\delta v}{\delta y} + \frac{1}{2} \left( \left( \frac{\delta u}{\delta y} \right)^2 + \left( \frac{\delta v}{\delta y} \right)^2 \right) \quad (4c)$$

$$\epsilon_{xy} = \frac{1}{2} \left( \frac{\delta u}{\delta y} + \frac{\delta v}{\delta x} + \frac{\delta u}{\delta x} \frac{\delta u}{\delta y} + \frac{\delta v}{\delta x} \frac{\delta v}{\delta y} \right) \quad (4d)$$

where,  $\epsilon_{xx}$ ,  $\epsilon_{yy}$  and  $\epsilon_{xy}$  are the Lagrangian finite strains and  $\frac{\delta u}{\delta x}$ ,  $\frac{\delta v}{\delta x}$ ,  $\frac{\delta u}{\delta y}$  and  $\frac{\delta v}{\delta y}$  are displacement gradients. The displacement gradients are calculated using the `NumPy` library functions for numerical differentiation. Then, the strain in  $x$  and  $y$  directions are visualized using the `matplotlib` library functions.

### 2.2.5. Exporting raw data

Along with full field strain visualization, correlated displacement and strain raw data can be obtained as CSV (Comma-Separated Values) files that facilitate subsequent visualizations using any other tool. To

implement this, the value of `export_RAW` in `DIC_Run` function must be changed to `True`, from the initial value of `False` which ensures a faster computational speed beforehand.

### 3. Illustrative example

RealPi2dDIC can be used to measure displacement and strain field of any planar surface of a structure during loading. Here we demonstrate, as an illustrative example, in-situ strain field measurement of a displacement controlled quasi-static test of a flat Aluminum dog-bone shaped specimen. The schematic diagram of the experimental setup is shown in Fig. 4. The specimen surface should have a good speckle-like pattern which is mandatory for optimal performance of DIC analysis [37].

The user has to install all dependencies following the instructions in the `readme.md` file supplied with the software package. The software can be run from the Raspberry Pi command line:

```
$ python RealPi2dDIC.py
```

The interface will ask for a directory name where the captured images and result files will be stored. Then the user has to select the image capture parameters based on the lighting conditions followed by the selection of ROI where DIC analysis would be done. Afterwards, RealPi2dDIC starts capturing images along with DIC analysis, and visualizes the results. Fig. 5. shows the longitudinal and transverse true strain field of the test specimen at different times of the test. It is observed that the failure has occurred where the strain was observed maximum. To validate the results, the raw images of this test has been analyzed with an open source DIC software DICE [17]. From Fig. A.6 it is discerned that maximum longitudinal strain is 22.5% and maximum transverse strain is 6.2%, both at point 4 which validates the strain data (Fig. 5.) acquired in-situ by RealPi2dDIC. A statistical analysis has also been done on the results of RealPi2dDIC and DICE for the same image set and it has been found that the results of RealPi2dDIC well follow the trend compared to the results of DICE (Table 2).

### 4. Impact

Digital Image Correlation is a robust tool widely applied in both academia and industry for its ease of use in complex operations without compromising the accuracy of measurement. RealPi2dDIC advances this

technology by introducing a very low-cost approach to monitor in-situ full field deformation and strain of structures under loading. This can be applied to a wide range of research fields and real-world problems where the assessment of deformation of structures is of interest. The application of this tool includes but is not limited to:

- Experimental solid mechanics (e.g. mechanical testing and characterization of metallic, polymeric, and composite materials)
- Study and monitoring of engineering components under mechanical or thermal loading
- Biomechanical applications (e.g. characterization of soft and hard tissue, etc.)
- Monitoring deformations in rail and road bridges
- Crack opening in civil engineering structures, especially in nuclear industry

This also has the potential to be the viable alternative of manual inspection techniques of large civil engineering structures like buildings, bridges, and power generation infrastructures.

RealPi2dDIC has been developed with a high emphasis on the simplification of input parameters through an interactive UI. Usage of Raspberry Pi and Pi camera offers a cost-effective solution for research groups and small-to-large industry where commercial systems are not available. This tool can be used in any indoor and outdoor environment because of the form factor of the system and can be controlled remotely.

The vast acceptance of Python3 as a programming language makes it the perfect environment for this tool, enabling continuous development of the code and adding new features that can help the researchers in an unprecedented way. Not to mention, the open nature of this tool also motivates the users to adopt, modify and extend the code to serve their purpose. This Raspberry Pi and Pi camera-based system can also be replaced with other powerful single-board computer systems (e.g. NVidia Jetson, Intel NUC, etc.) and imaging devices (e.g. Digital SLR cameras, etc.) to increase the computing efficiency at the cost of slightly higher economic investment. Furthermore, RealPi2dDIC is distributed under the GNU-GPLv3 license which makes commercial adoption of this tool readily available.

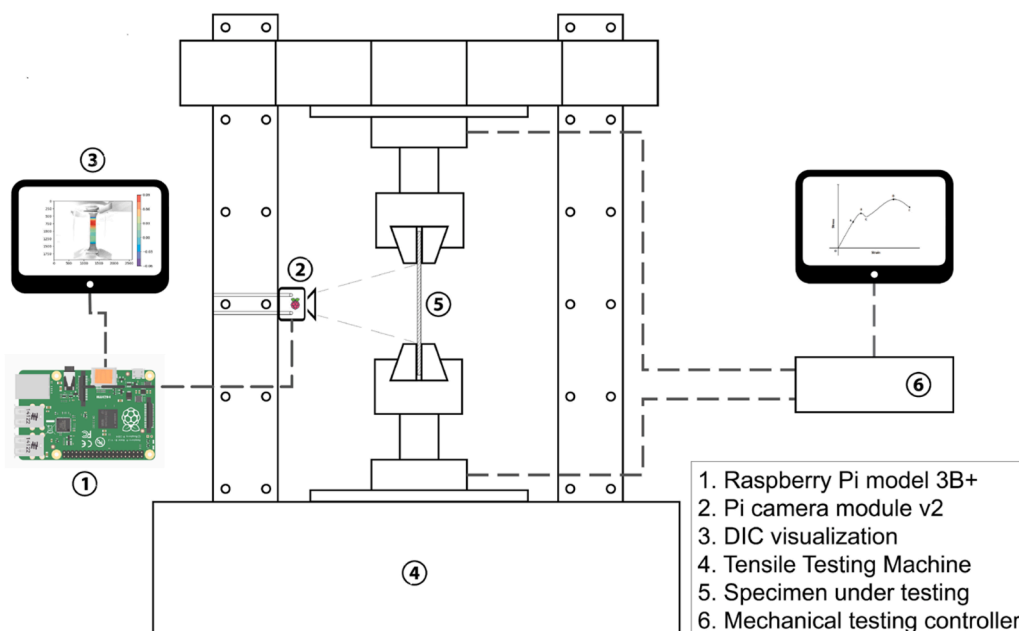


Fig. 4. Schematic diagram of a quasi-static test setup coupled with RealPi2dDIC system to monitor in-situ strain field.

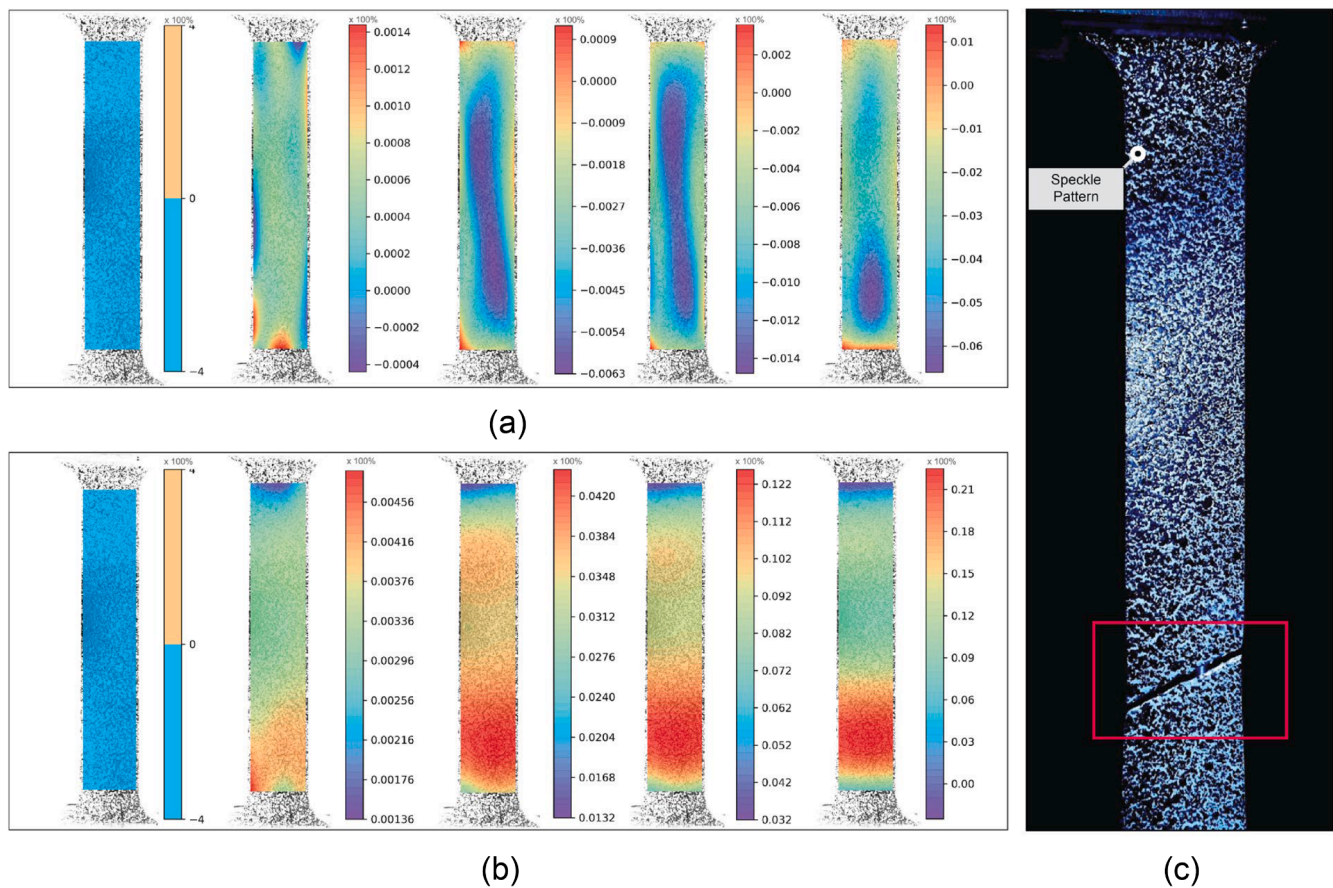


Fig. 5. In-situ true strain field of a specimen at different time of a quasi-static test (a. Transverse strain, b. Longitudinal strain), c. Specimen breaks at the place where the strain is maximum.

Table 2  
Statistical data comparison of the DIC results from RealPi2dDIC and DICe.

Statistical Parameter	RealPi2dDIC		DICe	
	Transverse	Longitudinal	Transverse	Longitudinal
Mean	-0.033	0.128	-0.039	0.118
Standard Deviation (SD)	0.014	0.053	0.014	0.051
Standard Error of Mean (SEM)	0.00041	0.00152	0.00043	0.00151

### 5. Conclusions

RealPi2dDIC is an open-source software to monitor in-situ full field deformation and strain of structures using local digital image correlation technique. For image correlation, computer vision-based Lucas-Kanade algorithm is used and B-spline interpolation is used for deformation field discretization, which reduces the computational cost. Unlike commercial CCD camera and typical desktop computer setups, this Raspberry Pi and Pi camera-based setup is more compact and reduces the cost significantly. The user-friendly UI allows the users to control camera and input parameters and makes it usable in various applications. RealPi2dDIC is coded in Python3 which makes it accessible to a wide user group motivating tweaking with the code to enable new features.

#### CRedit authorship contribution statement

**Partha Pratim Das:** Conceptualization, Formal analysis,

Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Muthu Ram Prabhu Elenchezian:** Conceptualization, Validation, Visualization, Writing – review & editing. **Vamsee Vadlamudi:** Methodology, Project administration, Supervision, Writing – review & editing. **Kenneth Reifsnider:** Methodology, Project administration, Supervision, Writing – review & editing. **Damien André:** Conceptualization, Software, Writing – review & editing. **Rassel Raihan:** Project administration, Resources, Supervision, Writing – review & editing.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

No data was used for the research described in the article.

#### Acknowledgements

The authors thank the researchers and staff of the Institute of Predictive Performance Methodologies and University of Texas at Arlington Research Institute for their support from developing the idea to publishing. The authors thank Shaikh Al Mahmud Bhuiyan for useful discussions and manuscript proofreading.

## Appendix

The raw images captured while the quasi-static test of the Aluminum specimen described in “Illustrative Example” have been analyzed using open source DIC software DICe and the results just before the failure are shown in Fig. A.6.

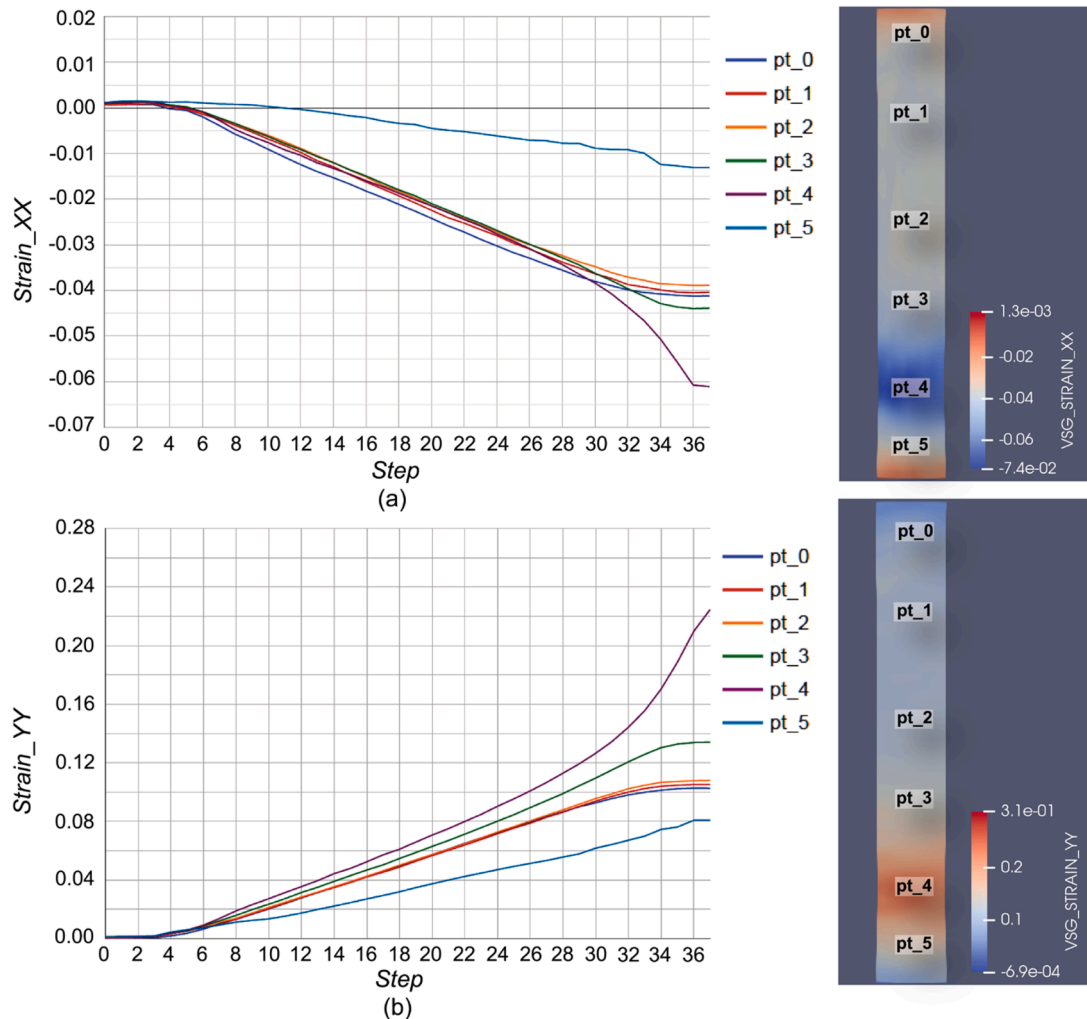


Fig. A.6. a. Transverse, and b. longitudinal true strain of six different points and the strain field of the specimen analyzed with DICe.

Among the 6 points under consideration, maximum transverse and longitudinal strain is observed at point 4 which resembles the data acquired by RealPi2dDIC.

## References

- [1] Wilson EJ. Chapter 17 strain-gage instrumentation. *Shock Vib Handb* 1988;11–7.
- [2] Motra HB, Hildebrand J, Dimmig-Osburg A. Assessment of strain measurement techniques to characterise mechanical properties of structural steel. *Eng Sci Technol Int J* 2014;17:260–9. <https://doi.org/10.1016/j.jestch.2014.07.006>.
- [3] Yang L, Wang Y, Lu R. Advanced optical methods for whole field displacement and strain measurement. In: 2010 Int. Symp. Optomechatronic Technol; 2010. p. 2–7. <https://doi.org/10.1109/ISOT.2010.5687394>. ISOT 2010.
- [4] Sirkis JS, Lim TJ. Displacement and strain measurement with automated grid methods. *Exp Mech* 1991;31:382–8. <https://doi.org/10.1007/BF02325997>.
- [5] Parks VJ. The moiré grid-analyzer method for strain analysis - discussion. *Exp Mech* 1966;6:287–8. <https://doi.org/10.1007/BF02327313>.
- [6] Pan B, Qian K, Xie H, Asundi A. Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. *Meas Sci Technol* 2009;20. <https://doi.org/10.1088/0957-0233/20/6/062001>.
- [7] Lecomte D, Bossuyt S, Cooreman S, Sol H, Vantomme J. Study and generation of optimal speckle patterns for DIC. *Proc SEM Annu Conf Expo Exp Appl Mech* 2007; 3:1643–9. 2007.
- [8] Sun Y, Pang JHL, Wong CK, Su F. Finite element formulation for a digital image correlation method. *Appl Opt* 2005;44:7357–63. <https://doi.org/10.1364/AO.44.007357>.
- [9] Bruck HA, McNeill SR, Sutton MA, Peters WH. Digital image correlation using Newton-Raphson method of partial differential correction. *Exp Mech* 1989;29: 261–7. <https://doi.org/10.1007/BF02321405>.
- [10] Chen J, Zhang X, Zhan N, Hu X. Deformation measurement across crack using two-step extended digital image correlation method. *Opt Lasers Eng* 2010;48:1126–31. <https://doi.org/10.1016/j.optlaseng.2009.12.017>.
- [11] Vend Roux G, Knauss WG. Submicron deformation field measurements: part 2. Improved digital image correlation. *Exp Mech* 1998;38:86–92. <https://doi.org/10.1007/BF02321649>.
- [12] Pan B, Xie HM, Xu BQ, Dai FL. Performance of sub-pixel registration algorithms in digital image correlation. *Meas Sci Technol* 2006;17:1615–21. <https://doi.org/10.1088/0957-0233/17/6/045>.
- [13] Tao G, Xia Z. A non-contact real-time strain measurement and control system for multiaxial cyclic/fatigue tests of polymer materials by digital image correlation method. *Polym Test* 2005;24:844–55. <https://doi.org/10.1016/j.polymertesting.2005.06.013>.
- [14] Wu R, Kong C, Li K, Zhang D. Real-time digital image correlation for dynamic strain measurement. *Exp Mech* 2016;56:833–43. <https://doi.org/10.1007/s11340-016-0133-6>.
- [15] Blaber J, Adair B, Antoniou A. Ncorr: open-source 2D digital image correlation Matlab software. *Exp Mech* 2015;55:1105–22. <https://doi.org/10.1007/s11340-015-0009-1>.



- [16] Yang J, Bhattacharya K. Augmented lagrangian digital image correlation. *Exp Mech* 2018. <https://doi.org/10.1007/s11340-018-00457-0>.
- [17] D. Turner, P. Crozier, P. Reu, Digital image correlation engine (DICE) reference manual, (2015). <https://www.osti.gov/servlets/purl/1245432>.
- [18] Belloni V, Ravanelli R, Nascetti A, Di Rita M, Mattei D, Crespi M. Py2dic: a new free and open source software for displacement and strain measurements in the field of experimental mechanics. *Sensors (Switzerland)* 2019;19:1–19. <https://doi.org/10.3390/s19183832>.
- [19] D. Andre, pydic, (n.d.). <https://gitlab.com/damien.andre/pydic>.
- [20] Olufsen SN, Andersen ME, Fagerholt E.  $\mu$ DIC: an open-source toolkit for digital image correlation. *SoftwareX* 2020;11:100391. <https://doi.org/10.1016/j.softx.2019.100391>.
- [21] R. Seghir, J.-F. Witz, S. Courdert, YaDICs-digital image correlation 2/3D software, (2014).
- [22] J. Réthoré, Ufreckles, (2018).
- [23] Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. *IJCAI* 1981.
- [24] Tong W. Formulation of Lucas-Kanade digital image correlation algorithms for non-contact deformation measurements: a review. *Strain* 2013;49:313–34. <https://doi.org/10.1111/str.12039>.
- [25] Bradski G. *The OpenCV Library*, Dr. Dobb's. *J Softw Tools* 2000.
- [26] Oliphant TE. *A guide to numpy*. Trelgol Publishing USA; 2006.
- [27] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJarrod, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İlhan, Feng Y, Moore EW, Vand erPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P. S. 1. 0 contributors, SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat Methods* 2020. <https://doi.org/10.1038/s41592-019-0686-2>.
- [28] D. Jones, picamera 1.13 documentation, Picamera. (2013). <https://picamera.re.adthedocs.io/en/release-1.13/index.html> (accessed August 20, 2020).
- [29] Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng* 2007;9:90–5. <https://doi.org/10.1109/MCSE.2007.55>.
- [30] Baker S, Matthews I. Lucas-Kanade 20 years on: a unifying framework. *Int J Comput Vis* 2004;56:221–55. <https://doi.org/10.1023/B:VISI.0000011205.11775.fd>.
- [31] Bouguet J-Y. *Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm*. Intel Corp 2001;5:4.
- [32] Tamgade SN, Bora VR. Motion vector estimation of video image by pyramidal implementation of Lucas Kanade optical flow. In: 2009 S Int. Conf. Emerg. Trends Eng. Technol; 2009. p. 914–7. <https://doi.org/10.1109/ICETET.2009.154>.
- [33] Pan B, Xie H, Wang Z, Qian K, Wang Z. Study on subset size selection in digital image correlation for speckle patterns. *Opt Express* 2008;16:7037–48. <https://doi.org/10.1364/OE.16.007037>.
- [34] Dierckx P. *An algorithm for surface-fitting with spline functions*. *IMA J Numer Anal* 1981;1:267–83.
- [35] Vermeij T, Hoefnagels JPM. A consistent full-field integrated DIC framework for HR-EBSD. *Ultramicroscopy* 2018;191:44–50. <https://doi.org/10.1016/j.ultramic.2018.05.001>.
- [36] Huang Y, Ji J, Lee K-M. Model-based digital image correlation for noncontact deformation measurement of strain field and mechanical property. *IEEE Trans. Ind. Informatics*. 2019;15:5109–18. <https://doi.org/10.1109/tii.2019.2896167>.
- [37] Dong YL, Pan B. A review of speckle pattern fabrication and assessment for digital image correlation. *Exp Mech* 2017;57:1161–81. <https://doi.org/10.1007/s11340-017-0283-1>.